2008

Real-Time Crop Row Image Reconstruction for Automatic Emerged Corn Plant Spacing Measurement

Lie Tang
Iowa State University, lietang@iastate.edu

Lei F. Tian
University of Illinois at Urbana-Champaign

Follow this and additional works at: http://lib.dr.iastate.edu/abe_eng_pubs
Part of the Agriculture Commons, and the Bioresource and Agricultural Engineering Commons

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/abe_eng_pubs/334. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.
REAL-TIME CROP ROW IMAGE RECONSTRUCTION FOR AUTOMATIC EMERGED CORN PLANT SPACING MEASUREMENT

L. Tang, L. F. Tian

ABSTRACT: In-field variations in corn plant spacing and population can lead to significant yield differences. To minimize these variations, seeds should be placed at a uniform spacing during planting. Since the ability to achieve this uniformity is directly related to planter performance, intensive field evaluations are vitally important prior to design of new planters and currently the designers have to rely on manually collected data that is very time consuming and subject to human errors. A machine vision-based emerged crop sensing system (ECSS) was developed to automate corn plant spacing measurement at early growth stages for planter design and testing engineers. This article documents the first part of the ECSS development, which was the real-time video frame mosaicking for crop row image reconstruction. Specifically, the mosaicking algorithm was based on a normalized correlation measure and was optimized to reduce the computational time and enhance the frame connection accuracy. This mosaicking algorithm was capable of reconstructing crop row images in real-time while the sampling platform was traveling at a velocity up to 1.21 m s⁻¹ (2.73 mph). The mosaicking accuracy of the ECSS was evaluated over three 40 to 50 m long crop rows. The ECSS achieved a mean distance measurement error ratio of -0.11% with a standard deviation of 0.74%.

Keywords. Corn plant spacing measurement, Image mosaicking, Machine vision, Real-time.

It is well known to corn producers that uneven plant spacing and emergence may reduce yield potential. Nielsen (2005) reported that uneven corn plant spacing within the row decreased yield up to two bushels per acre for every inch increase of standard deviation of plant-to-plant spacing. The performance of planter metering mechanisms directly determines how uniformly the seeds can be placed at an appropriate spacing. Consequently, planter performance will have substantial influence on crop establishment and final yield. Thus, the uniformity of plant spacing is an important parameter that planter manufacturers use to evaluate planter performance. Extensive field experiments thus are carried out by manufacturers during early crop growth stages and over different soil conditions. However, acquiring manual measurements of plant spacing requires substantial labor and time resources. Manual measurements not only limit the quantity of data, but also introduce human errors. This situation has made an automated and accurate early growth stage corn plant spacing sensing system highly desirable.

Developing the sensing technology to automatically identify each corn plant before harvesting has been a challenging problem in agricultural research. Machine vision technologies have been widely adopted to replace human labor in various inspection and measurement applications, and have been applied to these types of problems. Jia et al. (1991) and Shrestha and Steward (2003, 2005) investigated machine vision approaches for corn plant sensing.

To acquire plant spacing data through machine vision, the location of each individual plant needs to be determined. In addition, since a crop row is captured in a series of sequential images, i.e., video frames, the locations of plants need to be tracked or retained across these frames in order to calculate the plant spacing when two adjacent plants are separated by frames. There are two possible approaches to this problem: (1) plant shape feature-based tracking, and (2) soil background-based mosaicking.

In the plant shape feature-based matching approach, the locations of an individual plant in two sequential images are first found, and then those images are connected at these locations. This approach was adopted by Sanchez et al. (1996) for tracking cabbage plants in a series of sequential images. In the research reported herein in this current article, the first version of a corn plant sensing system was based on the plant feature-based tracking approach, but the system failed at field tests under two frequently occurring conditions: windy weather and large plant-to-plant gaps between two sequential frames. This failure occurred because the wind caused corn plant canopy movement between two sequential frames, thus making plants untrackable across the frames. Large inter-frame plant gaps could essentially break the tracking process as well when no plant appeared within the overlapped region of two sequential frames.

The video frame mosaicking approach relies on searching the spatial continuity of sequential frames in the image background consisting of soil and residue. In this way, two sequential frames are connected at a matched location found
in the image background, and wind interference and interframe plant gaps will not confound the process. Shrestha and Steward (2003) took this approach and developed a machine vision-based corn plant population sensing system. In their research, an image mosaicking algorithm was developed in which the minimum value of the sum of absolute errors over matching patches between two sequential frames was found. To improve the mosaicking speed, Shrestha et al. (2004) later proposed a gradient ascent method of the frame correlation surface with Kalman filter prediction and achieved an algorithm that was ten times faster than the minimum error method. They also indicated, however, that the success of this algorithm depended on precise shift prediction and the characteristics of the correlation surface.

Therefore, for the overall objective of developing a machine vision-based emerged crop sensing system (ECSS) for corn plant spacing measurement, the first task was to develop a mosaicking algorithm that can accurately and reliably retain the spatial information of a sequence of overlapped crop row video frames. Under this overall objective, one important requirement of the ECSS was real-time performance, i.e., the mosaicking process must keep up with the velocity of the image recording platform when sampling along a crop row since this will greatly enhance the system efficiency. For the ECSS, the targeted sampling speed was set about 1.11 m s⁻¹ (2.50 mph), which is similar to typical human walking speed. Another important requirement of the ECSS was the distance measurement accuracy. Ideally, planter engineers would require that the average corn plant spacing measurement error be smaller than ±5 mm. Typically, corn plant populations are around 30,000 plants per acre in the Midwestern U.S. With a 0.76 m (30 in.) row spacing, this population would result in a 0.18 m (6.97 in.) corn plant spacing. Thus, a ±5 mm distance error requires that the distance measurement error be about 3% or less of the true distance.

Consequently, the specific objectives of this research were to (1) develop a video frame mosaicking algorithm that is capable of accurately reconstructing corn plant row images in real-time, and (2) evaluate the performance of the mosaicking algorithm in terms its speed and accuracy.

**MATERIALS AND METHODS**

There were two main tasks in the crop row image reconstruction process. The first task was image acquisition and camera calibration, while the second task was video frame mosaicking.

**IMAGE ACQUISITION AND PREPROCESSING**

**Image Acquisition Under Outdoor Lighting Conditions**

To precisely measure corn plant spacing, machine vision-based measurement requires a spatial transformation that can relate image coordinates in the 2D image plane back into its original 3D world coordinate system. An inverse perspective transformation was derived by using a perspective transformation matrix technique described by Gonzalez and Woods (1992). In this calibration process, a unique calibration matrix can be determined from a minimum of four pixels. Because this calibration technique is a least-square based approach, a larger number of pixels is desired so that the matrix is over-determined for better

![Figure 1. Video recording using a modified bicycle under a sunny day.](image-url)
calibration accuracy. In the ECSS, the calibration algorithm was designed to take eight calibration points, which were manually selected from images of a calibration panel.

**IMAGE MOSAICKING**

**Correlation-Based Matching**

The image mosaicking algorithm was developed to compute interframe displacement so that two spatially overlapped sequential video frames acquired at different times could be connected to form a mosaicked image. A common approach for producing mosaicked images is to compare the linear relationship of intensity profiles in the neighborhood of potential matches using correlation measure as a similarity criterion (Forsyth and Ponce, 2002). This matching process involves a pixel-wise search, and the connecting point is given by the location of a matched connecting point. The output of the normalized correlation function ranges from -1 to +1, and it reaches its maximum value when the image intensity profiles of the two windows are related by an affine transformation \( A' = A\lambda + \mu \) for some constants \( \lambda \) and \( \mu \) with \( \lambda > 0 \). The invariance of \( C \) to affine transformations of the intensity function affords correlation-based matching techniques some degree of robustness over the intensity change across two sequential frames. For matching windows with zero mean and unit Frobenius norm, maximizing the correlation and minimizing the sum of squared differences (SSD) are equivalent. Mathematically, the correlation function calculates the cosine of the angle between vector \( \omega_n \) and vector \( \omega_n' \), whereas the SSD measures the 2-norm or the Euclidean length of vector \( \omega_n - \omega_n' \). The correlation and SSD measures for vectors \( \omega_n \) and \( \omega_n' \) are preferable to the sum of absolute difference (SAD) because SAD does not intrinsically measure the cross-correlation and the linear relationship between the vectors. In the normalized correlation-based method, an exact copy of the matched pattern can hardly be expected in the search area because some part of the pattern is usually corrupted by noise, geometric distortion, or occlusion. Therefore, instead of looking for an absolute match, it is more appropriate to seek the maximum of \( C \) over a search area in the current frame.

**Computational Optimization of Image Mosaicking**

Since the video equipment often moved along the row at a varying velocity, the size of the search area needed to be adequately large. The maximum allowable travel velocity is determined by the computational requirements of the image mosaicking algorithm and the computational capacity of the computing hardware. Obviously, a larger search area will substantially increase the computational time for a pixel-

\[
\omega = \frac{1}{p} \sum_{i=1}^{p} \omega_i \cdot \frac{1}{M} \end{equation}
\[
\omega' = \frac{1}{p} \sum_{i=1}^{p} \omega_i' \cdot \frac{1}{M} \end{equation}

The output of the normalized correlation function ranges from -1 to +1, and it reaches its maximum value when the image intensity profiles of the two windows are related by an affine transformation \( A' = A\lambda + \mu \) for some constants \( \lambda \) and \( \mu \) with \( \lambda > 0 \). The invariance of \( C \) to affine transformations of the intensity function affords correlation-based matching techniques some degree of robustness over the intensity change across two sequential frames. For matching windows with zero mean and unit Frobenius norm, maximizing the correlation and minimizing the sum of squared differences (SSD) are equivalent. Mathematically, the correlation function calculates the cosine of the angle between vector \( \omega_n \) and vector \( \omega_n' \), whereas the SSD measures the 2-norm or the Euclidean length of vector \( \omega_n - \omega_n' \). The correlation and SSD measures for vectors \( \omega_n \) and \( \omega_n' \) are preferable to the sum of absolute difference (SAD) because SAD does not intrinsically measure the cross-correlation and the linear relationship between the vectors. In the normalized correlation-based method, an exact copy of the matched pattern can hardly be expected in the search area because some part of the pattern is usually corrupted by noise, geometric distortion, or occlusion. Therefore, instead of looking for an absolute match, it is more appropriate to seek the maximum of \( C \) over a search area in the current frame.

**Computational Optimization of Image Mosaicking**

Since the video equipment often moved along the row at a varying velocity, the size of the search area needed to be adequately large. The maximum allowable travel velocity is determined by the computational requirements of the image mosaicking algorithm and the computational capacity of the computing hardware. Obviously, a larger search area will substantially increase the computational time for a pixel-
wise correlation-based matching algorithm. It is, thus, important to design a computationally efficient implementation to improve the travel velocity and to allow the use of a larger search area. This efficiency was realized through the use of recursion and lookup tables.

In a typical implementation of normalized correlation-based image mosaicking, one computes \( C(\vec{d}) \) (eq. 1), where \( \vec{d} \) is a 2-dimensional shift vector defined as \((d_u, d_v)\), at each pixel for each possible shift \( \vec{d} \) and finds the shift for which \( C(\vec{d}) \) is maximum. Although generating \( \omega_n \) and \( \bar{\omega} \) of the pre-selected matching window of size \( n \times n \) pixels in the previous frame only requires \( O(n^2) \) subtraction calculations, the search for a potential match in the current frame can be computational expensive because \( \omega_n \) and \( \bar{\omega} \) must be updated on a pixel-wise basis. If the size of the search area is \( m \times m \), the standard implementation of equation 1 requires \( O(m^2n^2) \) subtractions to obtain \( \omega_n \) and \( \bar{\omega} \) in order to cover every possible center positions of the window areas in the search area. However, it can be shown that:

\[
\left(\omega - \bar{\omega}\right)\left(\omega' - \bar{\omega}\right)=\omega \cdot \omega' - n^2 \cdot I \cdot 1'
\]  

(6)

where \( I \) and \( I' \) are two scalar variables representing the average pixel intensity values of windows \( W \) and \( W' \), respectively. Similarly, the following equation can be derived:

\[
\left\| \omega' - \bar{\omega} \right\|^2 = \sqrt{n^2 \cdot \omega' \cdot \omega' - \left(n \cdot I'\right)^2}
\]  

(7)

Using equations 6 and 7, the direct computation of vector \( \omega_n \), i.e., \( \omega' - \bar{\omega} \), is avoided, and \( I' \) can be computed recursively. To do this, \( I' \) is computed for a shift \( \vec{d} \) using the \( I_L \) of its immediate left neighbor matching window \( W_L \). More specifically, the contribution to \( I_L \) can be subtracted from the leftmost column of \( W_L \) and the contribution from the column immediately to the right of \( W_L \) can be added. This recursive process can be formulated as the equation below:

\[
I' = \frac{I_L \cdot n^2 - \sum_{i=1}^{n} W'_L(i,1) + \sum_{i=1}^{n} W'_L(i,n)}{n^2}
\]  

(8)

This recursive calculation can be applied both horizontally and vertically across the search area, which significantly reduces the required additions from \( O(m^2n^2) \) to \( O(2mn^2) \). Although the use of equations 6 to 8 can reduce the additions, \( O(m^2n^2) \) multiplication operations are still required to compute \( \omega \cdot \omega' \) and \( \omega' \cdot \omega' \). Since a single multiplication operation takes significantly more CPU clock cycles than a single assignment operation, the algorithm can be made even faster if all possible values of \( \omega \cdot \omega' \) and \( \omega' \cdot \omega' \) are pre-computed and stored in a 2-dimensional lookup table. Because the intensity values of pixels are represented by 8-bit numbers, the lookup table is easy to implement and only takes about 64K bytes of memory. During implementation, both standard and optimized mosaicking algorithms were coded, and tests were conducted to quantify the effects of this computational optimization on run time.

When the mosaicking algorithm was implemented, the memory required for the mosaicked image was pre-allocated according to the maximum available free memory. Preallocation saved time over dynamically reallocating memory for each pair of sequential video frames and helped maintain a constant frame processing time. This constant processing time was required to specify a maximum allowable travel velocity.

**IMPLEMENTATION OF FRAME TO FRAME MATCHING**

**Overall Setup of Frame-to-Frame Matching**

In the ECSS implementation, two spatially overlapped video frames were matched to find their spatial correspondence. In discussing the processing of the incoming video signal, the most recently acquired video frame is called the current frame, while the one immediately before it is called the previous frame. After each mosaicking operation using the previous and current frames, the current frame was copied into the memory containing the previous frame, and a new frame was acquired to replace the current frame in its memory location. Hence, the updating speed of the previous and current frames was determined by the operational time of mosaicking, not by the standard NTSC video frame rate (30 frames per second). Video frame elements used in the design of a correlation-based matching procedure for both previous and current frames are depicted in figure 3. Among those elements, the key element was the matching window area, which was first selected in the previous frame and then used as a matching window to find its best match (a matched window) in the current frame. Within the current frame, a search area was defined to be starting from and vertically symmetric around the mapped location of the matching window defined in the previous frame (fig. 3). The normalized correlation measure associated with every possible matched window within the search area was calculated. In addition, the matching window area selected from the previous frame was surrounded by a 5-pixel-wide vegetation buffer zone on all four sides, which in turn produced a larger wrapper, called a buffered-matching window. The vegetation buffer zone was used to better exclude vegetation from the matching process, as wind could possibly blow the adjacent vegetative objects such as leaves into the window area during the time interval between the acquisition of the previous and current frames, thus causing incorrect matching. The normalized correlation function (eq. 1) needs to be computed for all \((N - I/2 - 1) \times (M - H/2 - 1)\) locations of all possible matched windows within the search area to find the best match. There were two frame boundary buffer zones located at the upper and bottom edges of the 3rd quarter area of the previous frame, and each had a size of 160 \times 40 pixels. A matching window in the previous frame was selected from the 3rd quarter area. Because of the possibility of minor rotation and vertical displacement between sequential frames, the frame boundary buffer zones were excluded from the selection of the matching window in the previous frame to better ensure the existence of a match in the current frame.

Usually, image intensity values are used in correlation calculation. The camcorder used in this research had red, green, and blue signal bands generated by three CCD sensors. Although the intensity value at each pixel could be calculated by averaging its red, green, and blue values, a better approach was to directly use the red band signal for correlation calculation. Nitsch et al. (1991) indicated that soil and residue reflectivity curves had a constant to slightly
increasing trend over the 400 to 900 nm wavelength region. This spectral property implies that the red band signal is very likely to improve the texture information associated with image background (soil and residues) where the correlation-based match was sought. Therefore, the red color band was used to calculate the normalized correlation measure, which also improved the computational efficiency by eliminating pixel-wise intensity calculations.

Matching Window Selection
To ensure that a correlation-based matching algorithm performs well, the selection of a matching window in the previous frame is critical. Pratt (1974) stated two basic problems with the simple correlation measure. First, the broadness of the correlation function makes detection of the peak difficult since the simple correlation measure ignores the spatial relationship of points in the image. Second, image noise may lead to a false peak correlation. To alleviate these problems, a spatially matched filtering process or a statistical measure can be incorporated to decorrelate or “whiten” the image before the correlation-based matching process. In the case of objects in images, a whitening filter resembles a high-pass filter. Since images of natural scenes have a significant amount of natural spatial correlation, the statistical correlation measure utilizes edge outline comparisons between the two scenes. An edge detector type of spatial filter can increase the accuracy of correlation-based matching if the motion of the scene or the camera is purely translational. If a minor rotation does occur, high-pass filtering before correlation analysis can reduce the accuracy in finding the optimum peak. Although the ECSS had limited camera rotation, minor rotations did exist. So, in ECSS, high-pass filtering techniques were not implemented before the correlation analysis. Instead, the matching windows with the greatest high-frequency content and strongest textural information were used to correlate sequential frames (Trucco and Verri, 1998). Specifically, the pixel intensity variance within a matching window was calculated at each possible location by scanning through the 3rd quarter area of the previous frame. During this scanning process, a coarser step size (five rows per vertical step and 30 columns per horizontal step) was used to save time. Since wind can generate changes in vegetative object shapes across two matching frames, vegetation-free areas were prioritized in the process of matching window selection. Considering both strong texture and vegetation-free criteria, the following rules were adopted for the matching window selection process:

If there were vegetation-free matching window and buffered zone areas,

then the matching window was the window having the maximum variance and residing in a vegetation-free matching window and buffered zone area.

Else if there were vegetation-free matching windows,

then the matching window was the window having the maximum variance and without vegetation.

Else the matching window was the window having the maximum variance.

In practice, the above matching window selection process could robustly detect and select areas containing residues, rock edges, soil cracks, and rough soil surfaces. Consequently, the selected window areas provided more stable and distinguishable texture features for a more accurate inter-frame displacement calculation.

Figure 3. Relationship between frame area, search area, window area, vegetation buffer zone, and frame boundary zones (not to scale.)
Vegetation was detected using a color segmentation approach developed by Steward and Tian (1998). Their algorithm utilized an EGRBI (excess green, red-blue, intensity) color transformation, $K$-means clustering, and Bayes classification. A segmentation example is given in figure 4.

The performance of a correlation-based matching is also sensitive to the size of the matching window. Jain and Jain (1981) found that the accuracy of the area correlation method was poor when the window size was small. However, increasing the window size can substantially increase the computational time. Therefore, when determining the size of the matching window, a tradeoff is needed such that both the matching accuracy and the real-time mosaicking objectives can be achieved. In the ECSS, the matching window area size of $51 \times 11$ ($I = 51$ and $H = 11$) pixels was found to provide adequate textural information for a consistent mosaicking performance while meeting real-time requirements.

### Search Area Size Determination

In general, increasing the size of the search area allows a larger displacement between the previous and current frames. However, a larger search area also increases the time required for a frame connection to be made. In the ECSS, a larger search area is preferable because it allows frame segments of larger size to be connected into the mosaicked image, which means fewer frame connecting operations for a fixed length of crop row, thus reducing potential distance measurement errors due to possible imperfect mosaicking. On the other hand, implementing a larger search area inevitably increases the time needed for a frame connection operation, resulting in a greater change in viewing angle for the same object in two mosaicked frames, assuming a constant travel velocity. An angled view causes geometric distortions and object occlusions, which can potentially degrade the interframe window matching accuracy. Therefore, the size of the search area must be a compromised solution, and through trial and error, the size of the search area was defined to be $280 \times 40$ ($N = 280$ and $M = 40$) pixels.

### Mosaicked Image Generation

Once video interframe distance was found using the correlation measure, the video frame sequence was connected (fig. 5). When implementing this image mosaicking process, the currently acquired frame was matched with the previous frame. The distance increments along the $X$ and $Y$ axes after the $i$th mosaicking operation were:

$$\Delta x_i = x_{pi} - x_{ci}$$  \hspace{1cm} (9)$$

$$\Delta y_i = y_{ci} - y_{pi}$$  \hspace{1cm} (10)$$

where $(x_{pi}, y_{pi})$ is the position of the matching window center in the previous frame, and $(x_{ci}, y_{ci})$ is the position of the matched window center in the current frame.

The mosaicked image grew as image mosaicking proceeded. In the mosaicked image, the cumulative displacements at the $i$th image in the $X$ and $Y$ directions after $i$ mosaicking operations were:

$$X_i = \sum_{n=1}^{i} \Delta x_n + x_{pi}$$  \hspace{1cm} (11)$$

$$Y_i = \sum_{n=1}^{i} \Delta y_n$$  \hspace{1cm} (12)$$

Sequential frames were connected with detected translational motion parameters $\{\Delta x_i, \Delta y_i, X_i, Y_i\}$ at every mosaicking point where the matching window in the previous frame was centered. When the absolute value of $Y_i$ became large, corn plants could be shifted out from either the upper or lower edge of the mosaicked image. In other words, the displacement along the lateral direction to the crop row, $Y$, could accumulate and eventually drive the scene out of the vertical view range (243 pixels) of the mosaicked image. This would eventually happen when either the angle between the crop row direction and the video recording course (a) or the camera initial orientation (b) was not zero (fig. 5). The shaded areas from every frame in figure 5 constituted the actual scene being connected into a mosaicked image.

Assuming that a crop row is straight and that it originated from a point on the horizontal middle line of the first frame,
and defining the height of the frames as $H$ and the length of the mosaicked image at frame $i$ to be $L_i$, the constraint for preventing the crop row from vanishing from the mosaicked image at frame $i$ was:

$$L_i \cdot \tan(\max\{|\alpha - \beta|, |\beta|\}) < \frac{H}{2} \quad (13)$$

In the actual implementation of the image mosaicking procedure, the following rule was enforced to prevent crop row from vanishing:

If $|Y_i| > 40$ pixels, then $|Y_i| = |Y_i| - 40$ pixels \quad (14)

where the number 40 was the maximum allowable offset on the $Y$-axis.

When this rule was satisfied, a discontinuity occurred in the mosaicked image. These types of discontinuities were called mosaicking breakpoints. A mosaicking breakpoint could potentially split a corn plant, which would make the automated plant identification for spacing measurement more difficult. This problem was later solved in the plant identification algorithm.

A mosaicked image consisted of many fragments from a sequence of image frames. The connecting points of these fragments were called mosaicking points. Since these fragments originated from different portions of original frames, their positions relative to the sensing unit varied. To precisely compute the spacing across these fragments in a mosaicked image, every fragment in the mosaicked image was marked with a mosaicking point or a mosaicking breakpoint along with its image coordinates in its original single frame image (fig. 6). In this way, the camera calibration matrix, which was useful to correct the nonlinear distortion caused by the lens, could remain valid and could be used for plant spacing calculations from the mosaicked image.

**MOSAICKING PERFORMANCE TESTS**

First, real-time performance of the image mosaicking algorithm was investigated by calculating the mean processing time of a mosaicking operation over 100 mosaicking operations. This time was used to estimate the allowable travel velocity of the video recording platform. Mosaicking accuracy was tested by using the video tapes recorded in experimental fields in different states including Texas, Kansas, Illinois, and Iowa. The algorithm was repeatedly tested over sample paths about 40 to 50 m long. These sample paths had various weed infestation, soil tillage, and crop growth stage conditions. In particular, a careful validation test was conducted by using three crop row videos recorded in Texas and Iowa. The corn plants in these three sampled crop rows were mostly at V3 growth stage. The crop row in Texas had a length of 49.23 m, while the other two crop rows in Iowa were 40.98 m and 41.0 m long, respectively. In total, there were 12 video clips recorded from these crop rows; for every sampled crop row, there were two paths recorded in one direction and two paths recorded in the opposite direction. The true length of each tested crop row was manually measured by laying a measuring tape along the crop row. The camera was calibrated on-site immediately after the system was set up for recording. The total length measured from the mosaicked image was then compared with the true length to evaluate the accuracy of the mosaicking algorithm. Specifically, the distance measurement error ratio was calculated as:

![Figure 6. Sub-segment from mosaicked images of Texas cornfields, where “|” indicates mosaicking points, and “X” represents a mosaicking breakpoint.](image)
where ER, EL, and TL are the measurement error ratio, the estimated crop row length, and the tape-measured crop row length, respectively. Mean comparison of distance measurement error ratios from these three crops rows was conducted using one-way analysis of variance (ANOVA) provided by JMP 6 statistical software (SAS Institute, Inc., Cary, N.C.).

**RESULTS**

With a matching window size of 51 × 11 pixels and a search area of 280 × 40 pixels, the standard implementation of the normalized correlation-based matching algorithm required on average 0.43 s to connect two frames. Since the camera’s horizontal view was 762 mm (2.5 ft), the corresponding maximum allowable travel velocity (video recording speed) was 0.60 m s⁻¹ (1.34 mph). In contrast, the frame connection time was only 0.25 s after employing the recursive technique based on equations 6 to 8 and decreased further to 0.21 s when a lookup table was incorporated to calculate \( \omega \cdot \omega / C_0 \) and \( \omega / C_0 \cdot \omega / C_0 \) used in these equations. This improved computational process increased the overall mosaicking speed by 51%, and the maximum allowable video recording velocity was then correspondingly increased from 0.60 m s⁻¹ (1.34 mph) to 1.21 m s⁻¹ (2.73 mph).

Examples of mosaicked image sub-segments created by using the developed image mosaicking procedure are provided in figure 7. Through inspecting the continuity of objects at the mosaicking points, such as residues, plant leaves, and rocks, the image mosaicking algorithm was shown to work well visually. The notable uneven spacing of the mosaicking points in figure 7 was largely due to variable moving velocity of the sampling platform and the fact that every mosaicked fragment was connected at the center position of the matching window in the previous frame, whose location was determined by the matching window selection rules described earlier.

**Table 1. Mosaicking algorithm field test results.**

<table>
<thead>
<tr>
<th>Row Location</th>
<th>Err (m)</th>
<th>ER (%)</th>
<th>Mean (STD) of ER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texas</td>
<td>0.02</td>
<td>0.03</td>
<td>0.51 (0.34)</td>
</tr>
<tr>
<td></td>
<td>0.31</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>Iowa-1</td>
<td>-0.23</td>
<td>-0.57</td>
<td>-0.52 (0.40)</td>
</tr>
<tr>
<td></td>
<td>-0.11</td>
<td>-0.26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.44</td>
<td>-1.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.08</td>
<td>-0.19</td>
<td></td>
</tr>
<tr>
<td>Iowa-2</td>
<td>-0.38</td>
<td>-0.94</td>
<td>-0.30 (0.98)</td>
</tr>
<tr>
<td></td>
<td>-0.54</td>
<td>-1.34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.19</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.61</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>-0.11</td>
<td>0.74</td>
<td></td>
</tr>
</tbody>
</table>

When the mosaicking accuracy of the algorithm was tested over three sample crop rows, the algorithm achieved a mean crop row length measurement error ratio of -0.11% with a standard deviation (STD) of 0.74% (table 1). There was no evidence of significant differences in the error ratio across three test rows (\( F_{2,9} = 2.86, P = 0.11 \)), indicating that the mosaicking algorithm performed consistently over those experimental rows in spite of differences in soils and residue cover. However, the mean ER values from the Texas and Iowa rows were biased positively and negatively, respectively, implying that different camera calibrations most likely caused measurement error, since camera calibration error will result in a cumulative measurement error.

**CONCLUSIONS**

An algorithm for real-time corn crop row image reconstruction was developed and evaluated according to computational time and image mosaicking accuracy specifications. From this research, we can conclude:

- Crop row mosaicking can be done meeting the real-time requirements of a typical field data collection system. Specifically, the algorithm will allow a
maximum data collection speed of 4.4 km h⁻¹ (2.7 mph) when tested on a 400 MHz dual Pentium CPU computer.

- Crop row mosaicking accuracy under typical corn field conditions meets the requirements for corn planter performance testing. A mean mosaicking distance measurement error ratio of -0.11% with 0.74% standard deviation was observed in this research.

ACKNOWLEDGEMENT

This research of the Iowa Agriculture and Home Economics Experiment Station, Ames, Iowa, Project No. 0475, was supported by Hatch Act and State of Iowa funds. This research has also been supported by the Illinois Council of Food and Agricultural Research (C-FAR), project number IDACF 01-DS-3-1 AE and the University of Illinois. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Iowa State University and the University of Illinois.

REFERENCES


